A FRIENDLY AND FLEXIBLE HUMAN-ROBOT INTERFACE FOR CARL

L. Seabra Lopes, A. Teixeira, D. Gomes, C. Teixeira, J. Girão, N. Sénica

IEETA / Departamento de Electrónica e Telecomunicações

Universidade de Aveiro, 3810-193 Aveiro, Portugal

Abstract: This paper describes the architecture and current capabilities of Carl, a prototype of an intelligent service robot, designed having in mind such tasks as serving food in a reception or acting as a host in an organization. The approach that has been followed in the design of Carl is based on an explicit concern with the integration of the major dimensions of intelligence, namely Communication, Action, Reasoning and Learning. The paper focuses on the multi-modal human-robot communication capabilities of Carl, since these have been significantly improved during the last year.

Keywords: human-robot communication, natural language processing, touch screen interaction, animated face

1. INTRODUCTION

In recent years, robotics-related technologies have reached such a level of maturity that, now, researchers are feeling the next step is the development of personal robots, meaning, intelligent service robots capable of performing useful work in close cooperation/interaction with humans.

It will be necessary for robots of this new generation to comply with three criteria. First, these robots must be *animate*, meaning that they should respond to changing conditions in their environment. This requires a close coupling of perception and action.

Second, personal robots should be *adaptable* to different users and different physical environments. This typically requires reasoning and learning capabilities.

Finally, robots should be *accessible*, meaning that they should be able to explain their beliefs, motivations and intentions, and, at the same time, they should be easy to command and instruct.

In order to meet the *animate*, *adaptable* and *accessible* criteria for intelligent service robots, it is, therefore, necessary to include in their design such basic capabilities as linguistic communication, reasoning, reactivity and learning. "Integrated Intelligence" is an emerging keyword that identifies an approach to building intelligent artificial agents in which the integration of all those aspects of intelligence is considered (Seabra Lopes and Connell, 2001).

Given the progress obtained in sub-domains of AI and the maturity of the produced technologies, the "integrated intelligence" challenge seems to be the real challenge to face next. This is the focus of a national-funded project, CARL [1].

Artificial intelligence is often taken as a discipline aiming to develop artificial agents with a human level of intelligence. In the CARL project, we believe that it is more reasonable to develop useful robotic systems with hardware and intelligence tailored for specific applications. This will provide experience on how to integrate different technologies and execution capabilities and, eventually, will enable us to scale up to more general robot architectures.

This paper describes the current state of evolution of Carl, a prototype of an intelligent service robot

---

[1] "CARL - Communication, Action, Reasoning and Learning in Robotics", FCT PRAXIS/P/EEI/ 12121/1998.

developed by the project since 1999, which participated in the *AAAI Mobile Robot Competition and Exhibition* in 2001 and the *1st International Cleaning Robots Contest* in 2002.

Section 2 describes the hardware configuration and software architecture of the robot. Section 3 describes its global execution and interaction management system. Sections 4 and 5 describe the graphical and touch interaction capabilities. Section 6 describes the natural language processing capabilities. Section 7 concludes the paper with references to demonstration and ongoing work.

2. HARDWARE AND SOFTWARE ARCHITECTURE

Carl is based on a Pioneer 2-DX indoor platform from ActivMedia Robotics. It includes wheel encoders, front and rear bumper rings and front and rear sonar rings. The specific platform configuration also includes a micro-controller based on the Siemens C166 processor and an on-board computer based on a Pentium 266 MHz with PC104+ bus. The operating system is Linux RedHat 6.2. A Sony EVI D31 pan-tilt-zoom camera was added to enable such capabilities as object recognition and advanced navigation.

On top of the mobile platform, a fiber glass structure was added, which carries a Fujitsu-Siemens Lifebook laptop computer, based on an Intel Pentium III 700Mhz and running Linux Mandrake 8.0. The specific laptop model includes a touch screen to enable a touch interaction modality. Additionally, the fiber glass structure carries a VoiceTracker directional microphone array from Acoustic Magic, a speaker and a Creative WebCam Pro connected via USB port to the laptop.

Currently, Carl is 1.10 m tall. The microphone array is in a suitable position for speech recognition, since it is at a distance around 1 m from the mouth of the average adult speaker.

The fiber glass structure also includes a recipient for transporting small objects, equipped with an IR sensor for detecting the presence of objects. Finally, a set of 10 infra-red sensors have been attached to this structure in order to detect objects at different heights.

***Figure 1**: Current look of Carl*

The base computer and the laptop computer are connected by Ethernet cross-over cable and the robot has the possibility to be controlled and/or monitored by a 3rd computer via Wireless 802.11b WiFi card set on the laptop.

With this platform, we are developing an autonomous robot capable, not only of wandering around, but also of taking decisions, executing tasks and learning. The control and deliberation architecture of Carl

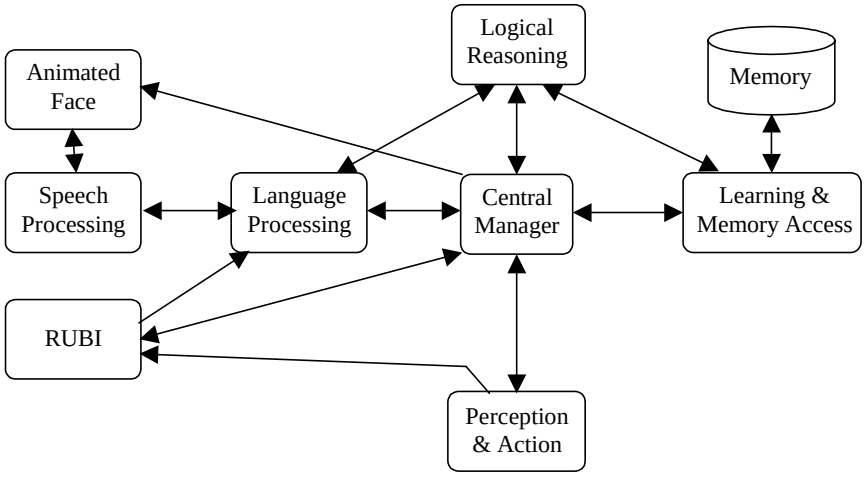

***Figure 2**: Current software architecture of Carl*

(Figure 2) reflects the goals of the project.

Human-robot communication is achieved through spoken and written language dialog as well as touch interactions. Speech processing is handled by a set of Linux processes, based on the Nuance speech tools for recognition and on IBM ViaVoice TTS for synthesis.

Touch screen interaction is controlled through the RUBI (for "Robot User Binding Interface") module, also implemented as a Linux process. Written language input can be captured through a virtual keyboard displayed on the touch screen. An animated face displays appropriate emotions.

Another Linux process handles general perception and action, including navigation. It is based on Saphira 8.1.8 and ARIA 1.1.8 API, the software interface for Pioneer robots.

High-level reasoning, including inductive and deductive inference, is mostly based on the Prolog inference engine (we use SWI-Prolog, a freeware version with a good C/C++ interface). Natural language parsing and generation is also implemented in Prolog. Another module of the architecture provides Carl with learning capabilities. A central manager coordinates the activities at the high level.

All computation is done on board. The perception and action process runs on the Pioneer base computer while all other processes run on the laptop computer. C++ and Prolog are the used programming languages.

The main advances with respect to previously published versions of Carl (Seabra Lopes, 2002; Seabra Lopes and Teixeira, 2000) are concerned with natural language processing, touch interaction and emotional display. These are described in special sections below.

3. EXECUTION AND INTERACTION MANAGEMENT

The central manager is an event-driven system. Events originating in the speech interface, in sensors or in the navigation activity as well as timeout events lead to state transitions. Such apparently different activities as dialog management and navigation management are integrated in a common unified framework.

```
state_transition(
  State,
  [no_biscuits],
  ( member(State,[explore,wander,stay]) ),
  nothing,
  [ retract_all_times,assert_go_to_refill_time,
    execute_task(go_to_refill_area) ],
  going_to_refill
).

state_transition(
  interacting,
  [ heard(tell(Phrase)) ],
  true,  % no restrictions
  acknowledge_told_fact(Phrase),
  [ execute_motion(stop),retract_all_times,
    memorize_told_fact(Phrase),assert_last_heard_time],
  interacting
).
```

***Figure 3**: Examples of state transitions*

It is mostly implemented in Prolog, in order to have easy access to the Prolog inference engine. Some parts of the manager are written in C language, either for reasons of efficiency or for access to the Linux inter-process communication facilities.

The central manager is essentially a state transition function (Figure 4) specified as a set of Prolog clauses. Each clause, specifying a transition, has a head of the following form:

```
state_transition(State,Events,Restrictions,
                                SpeechAct,Actions,NewState)
```

State is the current state; Events is a list of events that will cause a transition to NewState, provided that the Restrictions are satisfied. These events can be speech input events, navigation events, touch screen interface events, timing events, robot body events. SpeechAct, if not void, is some verbal message that the robot should emit in this transition. Actions are a list of other actions that robot should perform. These can be actions related to navigation, control of RUBI and the animated face, but also internal state update and dynamic grammar adaptation.

In total, Carl's state space includes around 15 states

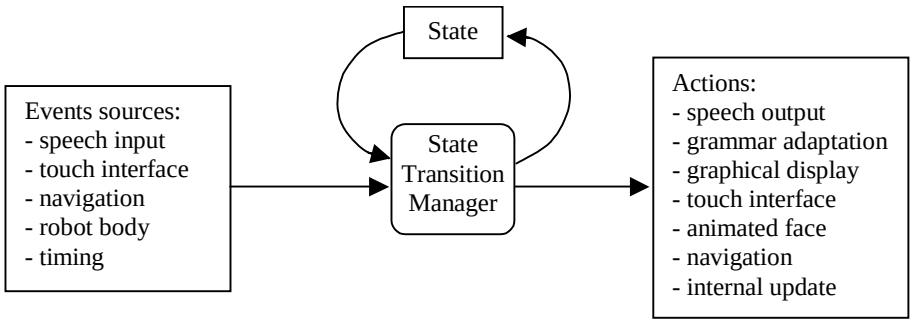

***Figure 4**: The central manager module - an event-driven process*

and 40 state transitions. Figure 3 shows two examples of state transitions. The first one is a transition from a normal motion state (explore or wander) or stay state to a state in which the main activity of the robot is to go to the refill area. The triggering event is the absence of biscuits in the food tray of the robot. This activity, event and state transition were introduced for the AAAI competition. The second state transition in Figure 3 is a transition to the same state, in this case the interacting state. The triggering event is the reception of an instance of the tell speech act. The robot immediately stops and acknowledges, then memorizes the told information. The time of this event is recorded, so that the robot may later recognize that the interaction is over, if it didn't finish with an explicit "good bye" from the human interactant.

4. GRAPHICAL AND TOUCH INTERFACE

In the previous configurations of Carl, the only available interaction modality was based on spoken language dialog (Seabra Lopes, 2002; Seabra Lopes and Teixeira, 2000). The touch screen facility, that comes with the Fujitsu-Siemens laptop computer recently installed in Carl, enables new interaction modalities. For that purpose, a graphical user interface, RUBI (for "Robot User Binding Interface"), was developed using QT Library, version 3.0. It allows the input of commands and information through touch as well as the display of monitoring and debug information. This way, the usability of the robot could be enhanced.

RUBI interacts with the following modules of the software architecture (see Figure 2.):

- Perception & Action - RUBI receives sensorial information from the Perception & Action module and displays it for monitoring purposes.
- Central Manager – RUBI receives internal state information from the manager and displays it for monitoring and debug purposes; user commands issued via RUBI are sent to the central manager for dispatching.
- Speech Processing – RUBI receives speech/face synchronization information from the speech synthesis process.
- Animated Face (see section 5.) - the received speech/face synchronization information is sent to the Animated Face process.

RUBI is organized into three areas. On the top-left corner, an animated face is shown. The face is visible at all times, and it's up to the user to maximize it to full screen, therefore hiding RUBI.

Below the animated face, a command panel is displayed. The following options are offered:

- You are ready: tell Carl that he is ready to do some task;
- Go to refill: tell Carl to go to the refill area, where food or something else can be fetched;
- Set refill area: tell Carl that he is currently at the refill area;
- Yes / No: provide a reply to a question of Carl;
- Change face texture: change texture of the animated face (see section 5.).
- Shutdown Carl: close all of Carl's software processes;

Most of these commands can also be issued via voice. However, given the current lack of robustness of the speech recognition technology in noisy environments, it can be more practical to use the touch interface in a particular situation.

To the right side of the face and the command menu, a large area, with three selectable folds, is displayed. (Figure 5). One of the folds is concerned with navigation. Here, motion commands (*move forward*, *move backward*, *turn left*, *turn right* and *stop*) are available through touch. These commands can also be issued through spoken language, but, again, given the particular situation, it can be more practical to use RUBI. Even from a safety point of view, the inclusion of these commands in RUBI would be mandatory.

This fold of RUBI displays monitoring information, allowing the user (as well as the developer) to accurately know at all times what is happening inside (something that could not be done before, since there was no graphical interface). The information that is currently being displayed includes:

- sensor values: infra-red sensors, sonars sensors, battery level;
- right and left wheel linear speeds;
- internal state of central manager;
- navigation mode;
- sentence recognized by the speech module.

Another fold of RUBI displays a virtual keyboard that can be actuated through touch and used for carrying out natural language dialogs with Carl (Figure 5, bottom). Although spoken dialog is our main aim, this can allow communication in environments were speech recognition is hard. This is useful also from a development point of view, since it allows to test the natural language understanding, dialog management and learning capabilities of Carl independently of the state of affairs at the speech recognition module.

The third fold of RUBI is a simplified explorer and viewer, enabling the user to see all the images saved in Carl's disk, as well as reorganizing the images directories. For future work, an online browser is being studied.

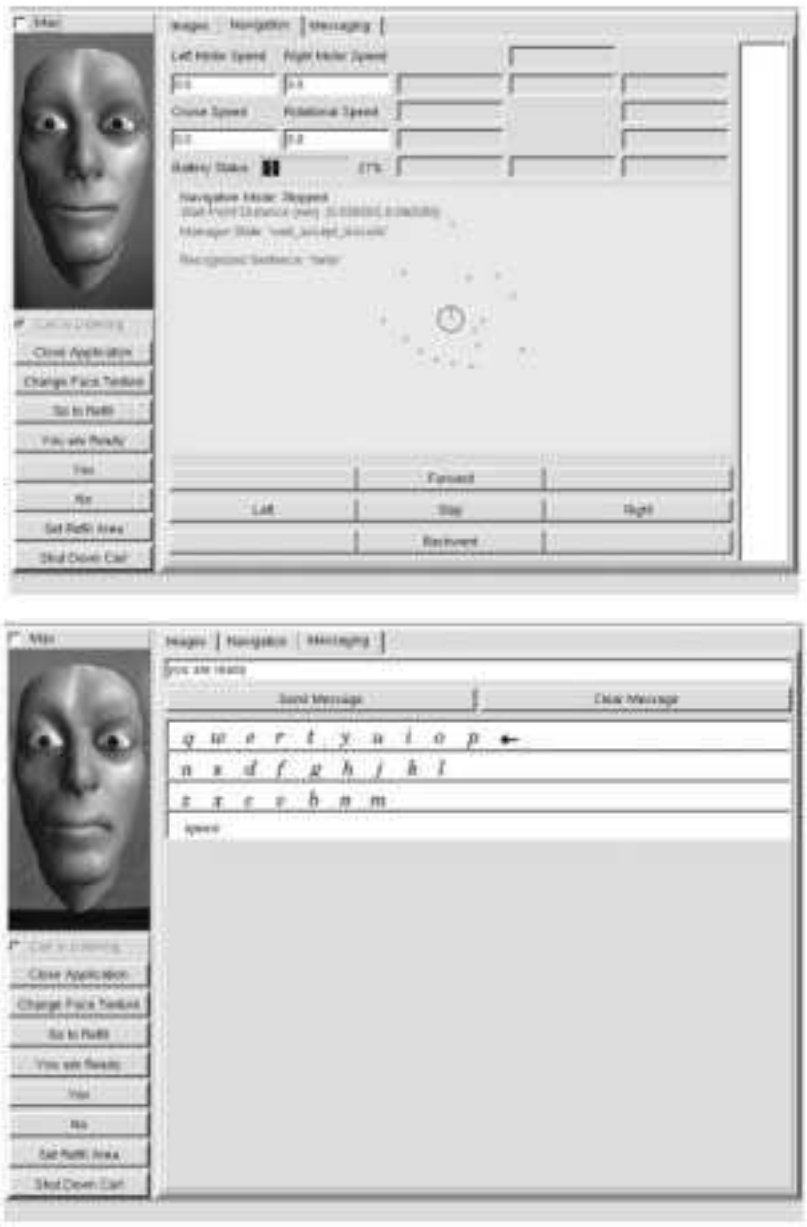

Figure 5 - Two aspects of the graphical and touch interface of Carl: navigation fold (top) and messaging fold (bottom)

5. ANIMATED FACE

The functionality of a robot, in terms of tasks that can be performed, is not all that matters. Many users will prefer robots that interact in a friendly way. We thought that an animated face might contribute to that.

The fact that the robot has a face is not exactly new, but we wanted a bit more than that. We wanted to create something that people could relate to. Taking this into consideration, the desirable characteristics were laid down. It had to be friendly and, at the same time, not too human, or people might reject the idea of a robot with a human face.

The muscle model approach to facial expression, developed by Keith Waters based on earlier work by Frederic Parke, is a good starting point (for general information on computer facial animation see the textbook of Parke and Waters, 1994).

The muscle model defines points to use as beginning and end for muscles. By moving one of these points we are actually creating a deformation caused by the stretch or contraction of a muscle. Using openGL and the C++ language, we were able to adapt the Parke module to our own set of points to draw a new skeleton for the face. The 3D model was built using only squares and triangles.

That done, we went in search of which emotions to express and the set of muscles we had to move to fulfil that objective. Mostly through a trial and error process, it was possible to combine a set of expressions that later became Carl's emotions.

The next step was to convert all this work into something believable. Carl couldn't just change emotion like a snapshot on a screen. Motion had to be added covering the interval points between muscle

Figure 6 - Carl's animated face with dynamic background

definitions. This gradual stretching and contraction of the muscles gave Carl a realistic behaviour.

Random small movements were also added to give some naturalness to the face. Such movements include blinking of an eye or a lip moving up and down from time to time. We believe this to somehow break the "general robot concept" which is something still and mechanical.

Finally, it would be nice to have a suggestive background for the face. If, instead of the face, the robot had a mechanical head, the background would be the space behind the head. Eventually, this led to the idea of capturing an image of the space behind Carl and use it as background for the animated face. A webcam is used for this purpose.

From the effect that Carl's face has on small children, who are actually delirious when they see Carl, we believe we were very successful in the objectives we set out to fulfil.

These natural human interfaces seem to be the next step. After we have machines that understand our natural ways, we'll want them to be able to express themselves in our natural ways as well. Part of this work is having the machine tell us how it feels.

6. NATURAL LANGUAGE PROCESSING

The goal of natural language processing is to extract semantics of natural language sentences and, conversely, to generate sentences from specifications of intended semantics.

Human-robot communication is one of the main research topics in this project. In the current configuration of Carl, natural language input can be received as a voice signal, captured via the VoiceTracker microphone array and processed by the NUANCE speech recognition software, or as written text, introduced via the virtual keyboard in the touch screen interface of Carl. Speech output is produced using IBM ViaVoice TTS.

The human-robot communication process is modelled as the exchange of messages, much like is done in multi-agent systems. The set of performatives or message types in our Human-Robot Communication Language (HRCL) is inspired in KQML (Seabra Lopes and Teixeira, 2000). Table I shows the currently supported set of performatives.

Table I – Currently supported performatives (S=sender, R=receiver)

Performative	*Description*
Register(S,R)	*S announces its presence to R*
Achieve(S,R,C)	*S asks R to perform action C in its physical environment*
Tell(S,R,C)	*S tells R that sentence C is true*
Ask(S,R,C)	*S asks R to provide one instantiation of sentence C*
Ask_if(S,R,C)	*S wants to know if R thinks sentence C is true*
Thanks(S,R)	*S expresses gratitude to R*
Bye(S,R)	*S says good-bye to R*
Dye(S,R)	*S (human master) asks R (robot) to close all execution processes*

Navigation guided by verbal user instructions has been demonstrated in the first phase of the project (Seabra Lopes and Teixeira, 2000). In a second phase, new functionalities were developed in order to support a larger grammar, eventually enabling Carl to parse and extract the semantics of 12000 different sentences (Seabra Lopes, 2002). Parsing was carried out by running an off-the-shelf C-based tool that returns a syntactic tree for a given sentence. From this tree, a semantic analysis program, developed by our group using Prolog, extracted a relational (first-order logic) representation of the semantics of the sentence. Dialog management and learning capabilities were also developed in this phase. In this version, Carl

participated in the *2001 AAAI Mobile Robot Competition and Exhibition* (Seattle).

Since then, the natural language processing module of Carl has been completely restructured. The current version is based on the *Attribute-Logic Engine* (ALE), a public domain logic programming and natural language processing system (Carpenter and Penn, 2001). The main features of ALE are:

- constraint logic programming system running over Prolog,
- use of typed feature structures,
- phase structure parsing,
- semantic-head-driven language generation,
- easy specification of subcategorization constraints, and
- easy specification of morphological rules.

Handling morphology and natural language generation were seen as major advantages of ALE, with respect to the tools previously used in Carl.

The compatibility with Prolog is another great advantage, since this is the language in which the high-level software modules of Carl are programmed. Logic program calls can be embedded in ALE grammars (as can be done in DCGs), thus allowing parsing to be interleaved with other system components. There is a port of ALE for SWI-Prolog, the specific Prolog implementation used in the project.

A typical grammar written in ALE starts with a "signature" section, which consists of a semantic hierarchy organizing the domain of objects involved in the definition of the grammar as well as in the domain of discourse. This semantic hierarchy is based on the specification of subtype relations, attributes for each type and inheritance of attributes from types to their descendents. After the signature, sections for lexicon, lexical rules (agreement, morphology) and grammar (phrase structure) rules should be included.

An implementation in ALE of the generation grammar used by Shieber et al. (1990) to illustrate the semantic-head-driven generation algorithm was used as starting point to develop the grammar for Carl. This implementation, included in the users guide of ALE (Carpenter and Penn, 2001, Appendix A3), contains only 14 words in its lexicon and 5 grammar rules. The top-level types in the signature section of this grammar are concerned with grammar categories, agreement in sentences, verb forms and semantics. The signature section is not well organized. A single type, pred, is used to denote different speech acts (declarations, commands), semantic relations, actions and subjects. The semantics type, sem, based on a relation name (of type pred) and the list of its arguments, is also limited in expressiveness. Therefore, besides expanding the language coverage of this grammar for Carl, it was necessary to largely restructure its signature section.

In the developed grammar, the root type (bot) and its immediate subtypes are as follows:

```
bot sub [ basicword, list, sem, form, agr,
          gram_cat, speech_act, gender ].
```

where:

- basicword is the type of the words representing different entities throughout the semantic hierarchy, including the natural language vocabulary; an analogy can be made with the atom type in Prolog; this roughly corresponds to the pred type in the original grammar.
- list is the root type for lists (comes from the original grammar)
- sem represents the semantics for sentences (see below).
- form and agr are concerned with verb forms and agreement (also comes from the original grammar).
- gram_cat is the root type for grammar categories; this subtree has been largely restructured and expanded with respect to the original grammar.
- speech_act is the root type for speech acts; includes those listed on Table I.
- gender (male, female and neutral) also used for agreement in sentences.

The subtree for sentence semantics is presented in Figure 7. Three main semantic subtypes are considered: objects (semobj), attributes of objects (sematt) and relations between objects (semrel). As relations often correspond to verbs, instances of the semrel type have as fields not only the relation name and objects, but also a field for a preposition and another for an adverb.

```
sem sub [ semrel, sematt, semobj, sem_yes_no, greating ].
  semobj sub [ ]
     intro [ obj:basicword, rels:sem_list ].
  sematt sub [ ]
     intro [ attname:basicword,  value:basicword ].
  semrel sub [ ]
     intro [  relname:basicword,
              obj1:semobj, obj2:semobj,
              prepname:basicword,
              vadverb: basicword
           ].
  sem_yes_no sub [ ] intro [bool: basicword].
  greating sub [ ].
```

Figure 7 - Type hierarchy for semantics

Several ALE macros were created to simplify the specification of lexicon entries. For example, one of the verb macros used in our grammar is:

```
verb(Verb) macro
   tverb,
   vform:nonfinite,
   vsubcat: [ (np,sem:Obj), (np,sem:Subj) ],
   sem: ( relname:Verb, obj1:Subj, obj2:Obj,
          prepname:none_, vadverb:none_ ).
```

This macro can be used to define transitive verbs, in non-finite form, subcategorized for a noun phrase as subject and another noun phrase as object. The semantics of the verb is a relation having the verb as name, the subject and object as arguments and having no associated preposition or adverb.

The last part of the grammar is mainly constituted by grammar (phrase structure) rules. For example, some verb phrases can be parsed by the following rule:

```
vp1 rule ( vp, form:Form,  subcat:Subcat, sem: Sem  )
===> sem_head> ( verb, vform:Form,
                    vsubcat: [ (np,sem:Obj) | Subcat ],
                    sem: Sem  ),
     cat> (np,sem:Obj).
```

The verbs acceptable by this rule are verbs that subcategorize a noun phrase as object, as it happens in the macro presented above. The vp1 phrase structure then fits into other rules until a complete sentence can be parsed. If the sentence is well formed, the parsing process directly delivers the semantics of the sentence, otherwise it fails.

After obtaining the typed feature structure representing the semantics of a given sentence, the last step is to convert it to a list of Prolog terms that can be asserted in the Prolog database (in tell speech acts) or matched with facts already existing in the database (as need in ask and ask_if speech acts). For instance, the semantics of the sentence "*peter is in the car of sandy*" would be given by the following list:

```
[ name_(X, peter), type_(Y, car), name_(Z, sandy),
  of(none_, none_, Y, Z) ].
```

This has a direct correspondence in first-order logic.

The current grammar has approximately 150 entries in the lexicon section and approximately 30 phrase structure rules.

As mentioned initially, one of advantages of ALE is its natural language generation capabilities. This means that, provided a description of the intended semantics, ALE can use the grammar to derive the corresponding sentence. Unfortunately, generation is an intrinsically non-deterministic process. What we observed was that, as new rules were added to the grammar, the generation process was becoming increasingly slower. For this reason, instead of using grammar-based generation, Carl continues to use a simpler template-based generation approach, introduced in the previous version of the language module.

7. CONCLUSION AND CURRENT WORK

A version of Carl, including the human-robot interaction capabilities described above, has been demonstrated at the welcome reception of IROS'2002 as part of the *1st International Cleaning Robots Contest* event (Lausanne, October 2002). Some pictures of this demonstration are available at the conference website (http://iros02.epfl.ch/gallery/view_album.php?set_albumName=Events). This demonstration was a great success, as it attracted a lot of public attention as well as media attention.

Current work is addressing the robustness of natural language processing for malformed sentences (common in the speech recognition domain). Some initial results are reported by Teixeira et al (2003). Another direction of work is concerned with computationally efficient natural language generation.

Another topic of interest for the group is robot learning from human interaction. Some recent results have been reported (Seabra Lopes and Wang, 2002).

REFERENCES

Carpenter, B. and G. Penn (2001) *ALE: The Attribute Logic Engine. User's Guide. Versio 3.2.1*, at http://www.cs.toronto.edu/~gpenn/ale.html, University of Toronto.

Labrou, Y. and T. Finin (1997) *A Proposal for a New KQML Specification*, University of Maryland at Baltimore County, technical report CS-97-03.

Parke, F.E. and K. Waters (1994) *Computer Facial Animation*, A.K. Peters Ltd.

Seabra Lopes, L. and A.J.S. Teixeira (2000) Human-Robot Interaction through Spoken Language Dialogue, *Proceedings IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, Japan, p. 528-534.

Seabra Lopes, L. and J.H. Connell, edrs. (2001) *Semisentient Robots* (special issue of *IEEE Intelligent Systems*, vol. 16, n. 5), Computer Society, p. 10-14.

Seabra Lopes, L. (2002) Carl: from Situated Activity to Language Level Interaction and Learning, *Proc. IEEE Int'l Conf. on Intelligent Robots and Systems (IROS'2002)*, Lausanne, Switzerland, p. 890-896.

Seabra Lopes, L. and Q.H. Wang (2002) Towards Grounded Human-Robot Communication, *Proc. 11th IEEE Int'l Workshop on Robot and Human Interactive Communication (ROMAN'2002)*, Berlin, Germany, p. 312-318.

Shieber, S.H., F.C.N. Pereira, G. van Noord and R.C. Moore (1990) «Semantic-Head-Driven Generation», *Computational Linguistics*, vol. 16 (1), p. 30-42.

Teixeira, A.J.S., L. Seabra Lopes, L. Ferreira, P. Soares and M. Rodrigues (2003) «Recent Developments on the Spoken Language Human-Robot Interface of the Robot Carl», submitted to *Robotica 2003 - Encontro Científico*.